Parle: parallelizing stochastic gradient descent

Pratik Chaudhari
Computer Science, UCLA
pratikac@ucla.edu

Carlo Baldassi
Data Science & Analytics, Bocconi University
carlo.baldassi@unibocconi.it

Riccardo Zecchina
Data Science & Analytics, Bocconi University
riccardo.zecchina@unibocconi.it

Stefano Soatto
Computer Science, UCLA
soatto@ucla.edu

Ameet Talwalkar
Machine Learning, CMU & Determined AI
talwalkar@cmu.edu

Adam Oberman
Mathematics & Statistics, McGill University
adam.oberman@mcgill.ca

ABSTRACT

We propose an algorithm called Parle for parallel training of deep networks that converges 2-4× faster than a data-parallel implementation of SGD, while achieving significantly improved error rates that are nearly state-of-the-art on several benchmarks including CIFAR-10 and CIFAR-100, without introducing any additional hyper-parameters. Parle exploits the phenomenon of wide minima that has been shown to improve generalization performance of deep networks and trains multiple "replicas" of a network that are coupled to each other using attractive potentials. It requires infrequent communication with the parameter server and is well-suited to single-machine-multi-GPU as well as distributed settings.

1 INTRODUCTION

The dramatic success of deep networks has fueled the growth of massive datasets, e.g. Google's JFT dataset has 100 million images, and even larger models. Parallel and distributed training of deep networks is paramount to tackle problems at this scale. Such escalation however hits a roadblock: stochastic gradient descent (SGD) with large batch sizes does not generalize well while small batch-sizes incur communication costs that quickly dwarf the benefits of parallelization. This paper presents an algorithm named Parle that:

(i) parallelizes the training of deep networks; it trains multiple copies, called "replicas", of the same model on possibly disjoint datasets over multiple GPUs while requiring very low communication bandwidth.
(ii) significantly improves upon convergence rate and generalization performance; it is 2-4× faster than data-parallel SGD while achieving nearly state-of-the-art validation errors.
(iii) insensitive to hyper-parameters; it does not introduce any extra hyper-parameters over SGD.

1.1 Approach

Trai ning a deep network involves solving the optimization problem $x^* = \arg\min_{x\in\mathbb{R}^N} f(x)$ where the weights are denoted by x and $f(x)$ is the average loss, say cross-entropy, over the entire dataset, along with a regularization term, say weight decay. We denote n copies, also called "replicas", of the weights by variables $x^1, \dots, x^n$. Consider the loss function of Elastic-SGD [1]:

$$x^* = \underset{x,\, x^1,\, \dots,\, x^n}{\arg\min} \; \frac{1}{n} \sum_{a=1}^{n} f(x^a) + \frac{1}{2\rho n} \, \|x^a - x\|^2; \tag{1}$$

where a parameter $\rho > 0$ couples two replicas x^a and x^b. The "reference" variable x which converges to the average of the replicas can be thought of as the master parameter server. Performing gradient descent on (1) involves communicating the replicas x^a for all $a \le n$ with the reference after each mini-batch update. Therefore, even though Elastic-SGD was introduced in the parallel setting, it nonetheless introduces significant communication bottlenecks.

We replace $f(x)$ by a smoother loss called local entropy [2, 3]

$$f_\gamma^\beta(x) := -\frac{1}{\beta} \log \left(G_{\gamma/\beta} \, * \, e^{-\beta f(x)} \right); \tag{2}$$

where $G_{\gamma/\beta}$ is the Gaussian kernel with variance γ/β. This loss function has been shown to bias SGD towards "wide minima" which generalize better than sharp ones [3]. Parle solves for

$$x^* = \underset{x,\, x^1,\, \dots,\, x^n}{\arg\min} \; \frac{1}{n} \sum_{a=1}^{n} f_\gamma^\beta(x^a) + \frac{1}{2\rho n} \, \|x^a - x\|^2. \tag{3}$$

If the stochastic gradient dynamics is ergodic [4, 5], the problem (1) is equivalent to minimizing (2). They differ, however, in their communication requirements: (2) is a non-distributed algorithm and does not involve any communication while Elastic-SGD (1) communicates frequently; Parle (3) strikes a balance between the two.

Remark 1 (Parle returns one single model). We let $\gamma, \, \rho \to 0$ as training progresses. This is motivated from connecting problem (3) with proximal point iteration where these parameters are step-sizes. Reducing the bandwidth of the Gaussian kernel and the coupling strength in (1) to zero forces different replicas to collapse together in the same region in the parameter space, even when they operate on disjoint datasets on a non-convex energy landscape. Thus, Parle maintains n replicas during training but returns one single model.

2 ALGORITHM AND ANALYSIS

The gradient of (2) can be written as

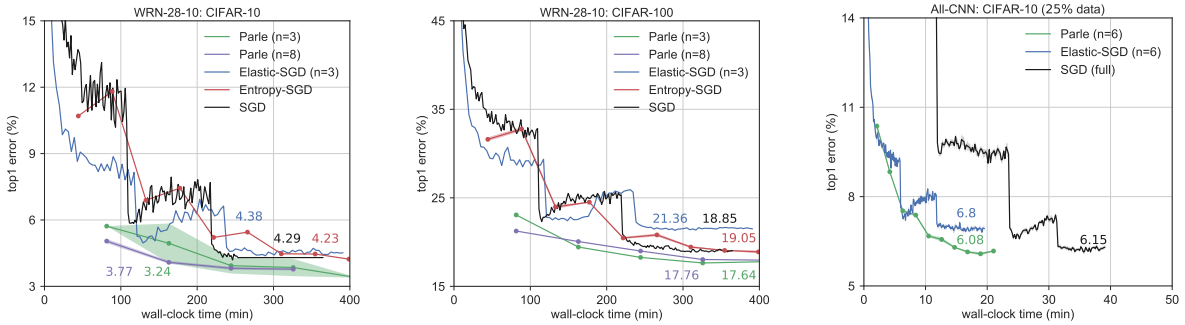
$$\nabla f_\gamma^\beta(x) = \gamma^{-1} \langle x - y \rangle \tag{4}$$

where the expectation $\langle \cdot \rangle$ is computed over a distribution

$$\mathbb{P}(y; x) \propto \exp\left(-\beta f(y) - \frac{\beta}{2\gamma} \|x - y\|^2 \right).$$

We use Langevin dynamics (SGLD) [8] to estimate this gradient and this involves a sequence of updates using the mini-batch gradient $f_b(y) + \gamma^{-1}(y - x)$ where b is the batch-size and a learning rate η. The thermal noise in SGLD is $\beta^{-1} = \frac{2\eta}{b}$.

The loss function of Elastic-SGD (1) can be minimized by synchronously computing the gradient of each replica, $\nabla f_b(x^a) + \rho^{-1}(x^a - x)$ and updating the master x after each iteration with the average of the replicas. This communication round introduces significant overheads for small batch-sizes or large networks.



(a) Wide-ResNet [6]: CIFAR-10 **(b) Wide-ResNet [6]: CIFAR-100** **(c) All-CNN-C [7]: CIFAR-10: 25% data**

Figure 1: Validation error of Parle (green, purple) compared with SGD (black), Elastic-SGD (blue) and Entropy-SGD (red): Figs. 1a and 1b show that Parle performs significantly better than these algorithms, both in terms of convergence rate as well as generalization. Fig. 1c shows that Parle with $n = 6$ replicas, each with only 25% of data is significantly faster and better than data-parallel SGD on the full dataset.

The loss function of Parle is motivated from the fact that Elastic-SGD with $\beta^{-1} = \frac{2\eta}{\delta}$ is equivalent to minimizing local entropy (2) if the gradient dynamics is ergodic; this was proved using stochastic homogenization in [4] and replica theory in [5]. The updates of the two algorithms can be interleaved in Parle to obtain:

$$y^a_{k+1} = y^a_k - \eta \nabla f(y^a_k) - \eta\gamma^{-1}\left(y^a_k - x^a_k\right), \tag{5a}$$

$$z^a_{k+1} = \alpha\, z^a_t + (1-\alpha)\, y^a_{k+1}, \tag{5b}$$

$$\text{if } k/\ell \text{ is an integer } \begin{cases} x^a_{k+1} &= (1-\tau)\, z^a_{k+1} + \tau\, x_k, \\ x_{k+1} &= \frac{1}{n}\sum_{a=1}^n x^a_k; \end{cases} \tag{5c}$$

here $\tau = \frac{\gamma}{\rho+\gamma}$ and η is the step-size. The number of steps of SGLD used to estimate the gradient $\nabla f^\beta_\gamma(x)$ are ℓ while α determines the the exponential averaging for the gradient estimate. We also add Nesterov's momentum to (5a) in our implementation.

We can show that as $\beta \to \infty$, minimizing (3) is equivalent to minizing the Moreau envelope [9, 10] of $f(x)$

$$f_{\gamma+\rho}(x) = \inf_y \left\{ f(y) + \frac{1}{2(\gamma+\rho)}\, \|x-y\|^2 \right\}$$

and the updates of Parle become, simply, the proximal point iteration [11] $x_{k+1} = \text{prox}_{(\gamma+\rho)f}(x_k)$. The parameters γ, ρ which are the bandwidth of the Gaussian kernel and the strength of the replica coupling are therefore seen as the step-sizes in PPI. They key idea in Parle is that $\text{prox}_{(\gamma+\rho)f}$ is split into $\text{prox}_{\gamma f}$ and $\text{prox}_{\rho f}$; the former is computed inexactly using SGLD updates while the latter is computed inexactly using Elastic-SGD updates.

Remark 2 (Parle is insensitive to hyper-parameters). The averaging parameter α is fixed to 0.75. PPI is insensitive to step-sizes and Parle begets the same property with respect to γ and ρ. We use an exponentially decreasing schedule for them of the form $\gamma_k = \gamma_0 \left(1 - \frac{1}{2B}\right)^{\lfloor k/\ell \rfloor}$ where B is the number of weight updates per epoch and γ_0 is fixed to 100. The schedule for η in Parle is set to be the same as that of SGD. The parameter ℓ determines the communication complexity since replicas are averaged every ℓ mini-batch updates. Using the non-asymptotic analysis of SGD [12] as a heuristic, it can be seen that the number of SGLD steps ℓ should scale linearly with $\eta\gamma$. Our current implementation sets $\ell = 25$. The values of all these hyper-parameters are fixed in our experiments, irrespective of the dataset or the network architecture.

Remark 3 (Communication requirements). Replicas in Parle synchronize weights with the master in step (5c) every ℓ mini-batch updates; steps (5a) and (5b) are executed independently. Since Parle is equivalent to Elastic-SGD if $\ell = 1, \gamma^{-1} = 0, \alpha = 0$, communication requirements of Parle are ℓ times smaller than those of Elastic-SGD.

3 EXPERIMENTAL RESULTS

Fig. 1 shows experimental results on benchmark datasets such as CIFAR-10 and CIFAR-100 [13] with standard data augmentation to match the setup of the baselines. Parle obtains nearly state-of-the-art errors with significant wall-clock time speedup; as a comparison, our generalization error is better than that of an ensemble of six DenseNet-100 networks [14].

Fig. 1c shows an example where Parle obtains better generalization as compared to SGD even when the latter operates on the full data. This is a direct consequence of setting $\rho \to 0$ whereby different replicas are forced to collapse to the same region in the parameter space in spite of working on different datasets.

Remark 4 (Federated learning). Our preliminary experiments show that the generalization performance of Parle is comparable to that of SGD even with $n \approx 1000$ replicas, each operating with 1% subset of the data. Coupled with ideas from the proximal operators literature and [15], this has implications for federated learning [16] on diverse computational platforms.

Remark 5 (Parle does not overfit on the training data). It is widely observed that state-of-the-art deep networks obtain good validation errors and near-zero training errors, i.e., the generalization gap is quite large. For instance, the training error for SGD in Figs. 1a and 1b is about 0.01%. Parle has a much lower generalization gap, it obtains a much higher training error: 4-5% for CIFAR-10 and 7-9% for CIFAR-100, in addition to a lower validation error. This suggests that minimizers found by Parle are qualitatively different from those of SGD: they lie higher in the energy landscape but still generalize well.

REFERENCES

[1] S. Zhang, A. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *NIPS*, 2015.

[2] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, "Local entropy as a measure for sampling solutions in constraint satisfaction problems," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 2, p. 023301, 2016.

[3] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, "Entropy-SGD: Biasing Gradient Descent Into Wide Valleys," *arXiv:1611.01838*, 2016.

[4] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and C. Guillame, "Deep Relaxation: partial differential equations for optimizing deep neural networks," *arXiv:1704.04932*, 2017.

[5] C. Baldassi, C. Borgs, J. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, "Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes," *PNAS*, vol. 113, no. 48, pp. E7655–E7662, 2016.

[6] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv:1605.07146*, 2016.

[7] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv:1412.6806*, 2014.

[8] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *ICML*, 2011.

[9] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.

[10] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, vol. 408. Springer, 2011.

[11] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[12] E. Moulines and F. R. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *NIPS*, 2011.

[13] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Computer Science, University of Toronto, 2009.

[14] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *arXiv:1608.06993*, 2016.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv:1602.05629*, 2016.